



**Predmet: Arhitektura računara**  
**Profesor: redovni profesor dr Dušan Regodić, dipl. inž.**

## T-4: Procesorski registri

---

1. INTERNI REGISTRI
2. PROGRAMSKI DOSTUPNI REGISTRI

CPU instrukcijski ciklus

## Interni registri

---

- uvedeni od strane projektanta procesora u zavisnosti od potreba koje primenjeni način projektovanja procesora zahteva
- pristup ovim registrima (upis/čitanje) programski nije moguć, već je to stvar ugrađenih algoritama po kojima se instrukcija izvršava
- služe za čuvanje sadržaja u različitim fazama izvršavanja neke instrukcije, s tim da se sadržaj može koristiti samo u okviru te instrukcije (a ne neke druge instrukcije)
- obuhvataju kontrolne i statusne registre
- pripadaju organizaciji računara

## Kontrolni registri (1)

---

□ pri izvršavanju instrukcije, za prenos podataka između memorije i procesora, obično se koriste sledeći kontrolni registri:

- **programski brojač** (*Program Counter*) – **PC**, sadrži adresu naredne instrukcije koji treba dohvatiti
- **instrukcijski registar** (*Instruction Register*) – **IR**, sadrži instrukciju koja je poslednja uzeta
- **memorijski adresni registar** (*Memory Address Register*) – **MAR**, sadrži adresu lokacije u memoriji
- **memorijski registar podataka** (*Memory Data Register*) – **MDR**, sadrži podatak koji treba upisati u memoriju ili je najskorije pročitano iz memorije

□ PC se inkrementira implicitno nakon svake instrukcije, ali se može postavljati i eksplicitno instrukcijama skoka (programski dostupan)

## Kontrolni registri (2)

---

- da bi bili procesirani, podaci se moraju dovesti na ulaze ALU
- ALU može da bude direktno priključena na MDR ili na programski dostupne registre
- takođe, ALU može da ima i posebne, bafer registre koji su priključeni na MDR ili programski dostupne registre

# Statusni registar (1)

---

statusni registar (*Program Status Word*) – PSW, sadrži uslovne kodove i neke statusne informacije

PSW se sastoji od određenog broja bitova, tj. indikatora koji se setuju nezavisno jedan od drugog

indikatori su podeljeni u dve grupe: **indikatori statusnog karaktera** i **indikatori upravljačkog karaktera**

indikatori statusnog karaktera se postavljaju hardverski na osnovu rezultata izvršavanja instrukcija, a proveravaju se instrukcijama uslovnog skoka

indikatori upravljačkog karaktera se postavljaju softverski tokom izvršavanja posebnih instrukcija, a proveravaju se hardverski

zbog indikatora upravljačkog karaktera, ovi registri su delom programski dostupni

## Statusni registar (2)

---

### Uobičajeni statusni indikatori PSW registra:

**N** – bit se postavlja na 1 ako je rezultat operacije negativan

**Z** – bit se postavlja na 1 ako je rezultat operacije nula

**C** – bit se postavlja na 1 ako ima prenosa/pozajmice pri aritmetičkim operacijama nad neoznačenim veličinama

**V** – bit se postavlja na 1 ako ima prekoračenja pri aritmetičkim operacijama nad celobrojnim veličinama sa znakom

### Uobičajeni upravljački indikator PSW registra:

**I** – bit se postavlja na 1 ako su dozvoljeni maskirajući prekidi

## Programski dostupni registri (1)

---

- sadržaj registara se može upisivati ili čitati programskim putem, tj. tokom izvršavanja instrukcija
- registar kome se pristupa se specificira:
  - eksplicitno, u okviru adresnog polja instrukcije
  - implicitno, u okviru koda operacije instrukcije
- služe za čuvanje sadržaja koji je rezultat izvršavanja neke instrukcije, s tim da se taj sadržaj kasnije može koristiti prilikom izvršavanja drugih instrukcija
- pripadaju arhitekturi računara

## Programski dostupni registri (2)

---

broj i uloga ovih registara razlikuje se od procesora do procesora

**Najčešće se koriste sledeći registri:**

1. registri podataka – DR (*Data Registers*)
2. adresni registri – AR (*Address Registers*)
3. registri opšte namene – GPR (*General-purpose Registers*)
4. programski brojač – PC (instrukcije skoka)
5. statusni registar – PSW (indikator upravljačkog karaktera)



## Registri podataka (1)

---

- omogućavaju brži pristup podacima zato što se tokom izvršavanja programa podaci uzimaju iz registara, a ne iz memorije (pristup memoriji je skoro za red veličine sporiji)
- koriste se u slučaju kada se tokom izračunavanja više puta koristi isti podatak
- međurezultati obrade se često smeštaju u registre podataka, kako bi im se kasnije brže pristupalo
- ukoliko je konačni rezultat u registru podataka, na kraju obrade se prebacuje u memoriju

## Registri podataka (2)

---

služe samo za čuvanje podataka, i ne mogu se koristiti pri računanjima adresa operanada

### Razlozi za uvođenje registara podataka:

ako se jadanput pristupilo nekom podatku, velika je verovarnoća da će se on uskoro opet koristiti

sekvencijana priroda računanja ima osobinu da se često rezultat jedne operacije koristi kao ulazni podatak za narednu operaciju

# Adresni registri

---

- omogućavaju brži pristup adresama zato što se tokom izvršavanja programa adrese uzimaju iz adresnih registara, a ne iz memorije (pristup memoriji je skoro za red veličine sporiji)
- mogu biti opšteg karaktera, ali mogu biti i specijalizovani u skladu sa primenjenim načinom (modom) adresiranja
- specijalizovani registri:
  - bazni registri – BR (*Base Registers*)
  - indeksni registri – XR (*Index Registers*)

## Registri opšte namene

---

mogu da imaju različite uloge, na primer da čuvaju podatke kao registari podataka ili adrese kao adresni registri

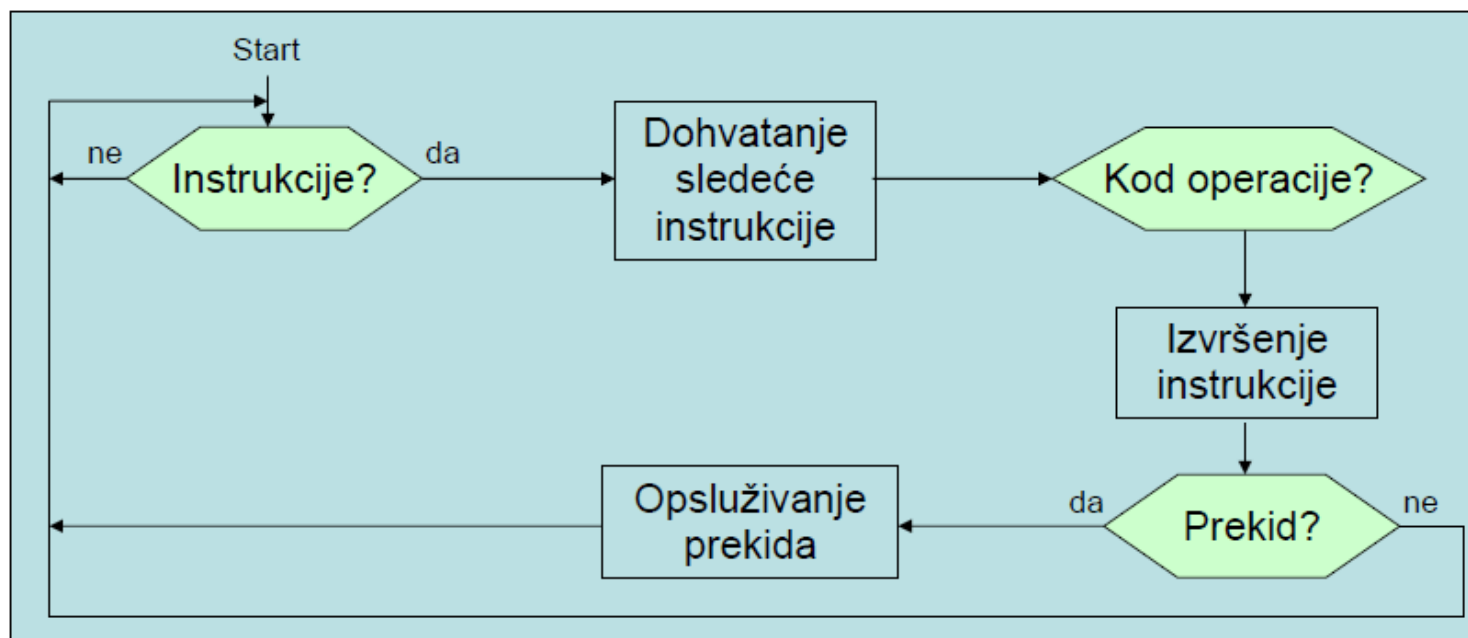
**Zavisno od načina projektovanja, postoje procesori koji:**

imaju specijalizovane registre (registre podataka, adresne registre, bazne registre i indeksne registre)

imaju samo registre opšte namene, koji preuzimaju uloge specijalizovanih registara

## CPU instrukcijski ciklus

- Procesor izvršava instrukcije smeštene u memoriji jednu za drugom dok ih ima.
- Nakon završetka izvršenja svake instrukcije, procesor proverava da li ima zahteva za prekidom.



## Dohvatanje instrukcije

---

Sekvenca mikro operacija koje se izvršavaju prilikom dohvatanja instrukcije iz memorije:

- sadržaj PC se prenosi u MAR
- inkrementira se sadržaj PC registra (ova operacija može da se radi paralelno sa pristupom memoriji)
- kao rezultat operacije čitanja memorije, instrukcija se smešta u MDR
- sadržaj MDR se prenosi u IR

# Izvršavanje instrukcije

---

Sekvenca mikro operacija koje se izvode prilikom izvršavanja instrukcije zavisi od same instrukcije.

**Primer 1: ADD R1, R2, R0**

$(R1) + (R2) \rightarrow R0$

- adrese registara R1, R2 i R0 se uzimaju iz formata instrukcije
- sadržaji registara R1 i R2 se dovode na ulaze ALU radi sabiranja
- izlaz iz ALU se prenosi u registar R0

**Primer 2: ADD X, R0**

$M(X) + (R0) \rightarrow R0$

- adresa mem.lok. X se uzima iz formata instrukcije i stavlja u MAR
- podatak pročitani iz memorije (sadržaj X) se smešta u MDR
- sadržaj MDR se dodaje na R0

# Opsluživanje prekida

---

Sekvenca mikro operacija koje se izvršavaju prilikom opsluživanja prekida:

- sadržaj PC se prenosi u MDR (da bi bio sačuvan)
- MAR se puni adresom na kojoj treba sačuvati sadržaj PC-a
- PC se puni adresom prve instrukcije u prekidnoj rutini
- sadržaj MDR (stara vrednost PC-a) se smešta u memoriju (adresa je u MAR)



**HVALA VAM NA  
PAŽNJI**

